

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right, containing the date. Below the banner, several thin, curved lines in dark blue and light grey originate from the left and sweep upwards and to the right.

8/13/2023

Basic Chatbot

SayabiDevs AI & Data Science Team

Authored By:

Muhammad Sabih Ur Rehman

Muhammad Anas Ansari

Ifra Siddiqui

Abdullah Hashim

TASK:

In this introductory project, interns will collaboratively work in groups of three to create a basic rule-based chatbot. The chatbot will be designed to interact with users and provide information on a specific topic. This task is aimed at familiarizing interns with the fundamentals of natural language processing (NLP) and basic AI development concepts.

Interns will gain a basic understanding of natural language processing concepts, including tokenization, entity recognition, and intent detection. Interns will apply their coding skills to develop a simple rule-based chatbot using a programming language of their choice (Python, Java, etc.). The chatbot should be able to engage in a text-based conversation with users, respond to common queries, and provide relevant information on the chosen topic. Interns will implement simple decision-making logic to handle different user inputs and respond appropriately. Interns will work collaboratively within their groups, utilizing version control (e.g., GitHub) to manage code and track changes.

Each group will select a specific topic for their chatbot (e.g., weather, trivia, jokes, etc.). Interns will gather a small dataset of sample user queries and possible responses related to their chosen topic. Design a basic conversation flowchart that outlines possible user inputs and corresponding bot responses. Write code to create the rule-based chatbot using the selected programming language. Implement basic NLP functions for tokenization and intent detection. Interns will create a simple user interface where users can input queries and interact with the chatbot. Test the chatbot with various user inputs and identify and fix any bugs or issues. Each group will provide a brief documentation outlining the chatbot's functionality, the chosen approach, and any challenges faced during development.

Rule-based chatbot code hosted on a version control platform (e.g., GitHub). User interface for interacting with the chatbot. Documentation describing the chatbot's functionality, development approach, and challenges faced.

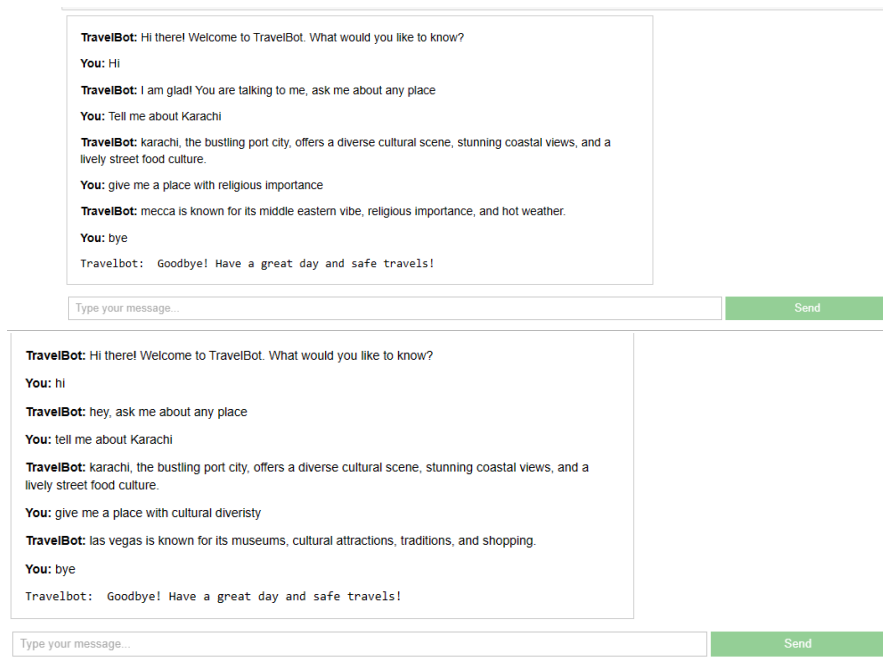
Interns will be evaluated based on the functionality and interactivity of their chatbot, the quality of their code, the clarity of their documentation, and their ability to work collaboratively as a team.

This project aims to provide interns with a hands-on experience in building a simple AI application and lay the foundation for more complex projects in the future.

TravelBot Documentation:

Project Overview:

This project is a chatbot that can answer tourism-related queries. The chatbot is trained on a dataset of tourism-related text, and it uses a technique called **cosine similarity** to find the most similar sentence in the dataset to the user's query. The chatbot then responds with the sentence that it found.



The image displays two screenshots of the TravelBot user interface. Each screenshot shows a chat window with a list of messages on the left and a text input field with a 'Send' button on the right. The messages in the chat window are as follows:

TravelBot: Hi there! Welcome to TravelBot. What would you like to know?
You: Hi
TravelBot: I am glad! You are talking to me, ask me about any place
You: Tell me about Karachi
TravelBot: karachi, the bustling port city, offers a diverse cultural scene, stunning coastal views, and a lively street food culture.
You: give me a place with religious importance
TravelBot: mecca is known for its middle eastern vibe, religious importance, and hot weather.
You: bye
TravelBot: Goodbye! Have a great day and safe travels!

The input field in both screenshots contains the text 'Type your message...' and the 'Send' button is green.

TravelBot UI

Flow of the Project:

The project flow is as follows:

1. Import libraries and load data.
2. Define response generation function.
3. Define greeting and farewell functions.
4. Create user interface.
5. Define message sending and receiving function.
6. Display interface.

Challenges Faced:

- **Finding a good dataset:** The dataset that I used for this chatbot is relatively small. This means that the chatbot may not be able to answer all tourism-related queries. I would like to find a larger dataset in the future to improve the chatbot's capabilities.
- **Adapting to Diversity:** The challenge of accommodating an array of user inputs, from cordial greetings to inquisitive questions, compelled a flexible approach that embraced the diverse ways travelers engage.
- **Improving the natural language processing capabilities:** The chatbot currently uses a simple technique called cosine similarity to find the most similar sentence in the dataset to the user's query. This technique is not always accurate, and it can sometimes misunderstand the user's query. I would like to improve the natural language processing capabilities of the chatbot in the future to make it more accurate and to reduce the chances of misunderstanding the user's query.

Limitations:

- The chatbot is only trained on a dataset of **tourism-related text**. This means that it may not be able to answer queries about other topics.
- TravelBot's responses are born from **exact keyword matches**, occasionally missing the subtleties of nuanced queries that defy categorization.
- The chatbot is **not perfect**. It may sometimes misunderstand the user's query or provide inaccurate information.
- The chatbot is **still under development**. There are many features that could be added to improve its functionality.
- The current design sets the stage for intimate dialogues, where TravelBot guides a solo traveler. The journey towards accommodating multiple voices and dynamic interactions beckons on the horizon.

Future Improvements:

The following are some future improvements that could be made to the chatbot:

- Use a **larger dataset** to improve the chatbot's accuracy.
- Improve the natural language **processing capabilities** of the chatbot to better understand the user's intent.
- Add **more features** to the user interface to make it more interactive and engaging.
- TravelBot aspires to become a **polyglot**, unraveling stories in myriad languages, connecting with travelers from every corner of the globe.
- Add a **mechanism** for the user to rate the chatbot's responses to improve the chatbot over time.

Flow Diagram:

The grand symphony of TravelBot's interaction unfolds in a captivating ballet, choreographed to the rhythm of user engagement:

